

Week 3

Using String Patterns, Ranges



Using String Patterns, Ranges, and Sets

- At the end of this lesson, you will be able to describe how to simplify a SELECT statement by using:
 - String patterns
 - Ranges, or
 - Sets of values



Retrieving rows from a table

```
db2 => select * from Book
```

Book_ID	Title	Edition	Year	Price	ISBN	Pages	Aisle	Description
B1	Getting started with DB2 Express-C	1	2010	24.99	978-0-98086283-1-1	300	DB-A02	Teaches you the fundamentals
B2	Database Fundamentals	1	2010	24.99	978-0-98666283-5-1	280	DB-A01	Teaches you the essentials of
B3	Getting started with DB2 App Dev	1	2011	35.99	978-0-98086283-4-1	345	DB-A03	Teaches you the essentials of
B4	Getting started with WAS CE	1	2010	49.99	978-0-98946283-3-1	458	DB-A04	Teaches you the essentials of

4 record(s) selected.

```
db2 => select book_id, title from Book
```

Book_ID	Title
B1	Getting started with DB2 Express-C
B2	Database Fundamentals
B3	Getting started with DB2 App Dev
B4	Getting started with WAS CE

4 record(s) selected.

```
db2 => select book_id, title from Book  
WHERE book_id='B1'
```

Book_ID	Title
B1	Getting started with DB2 Express-C

1 record(s) selected.

Retrieving rows - using a String Pattern

- WHERE requires a predicate
- A predicate is an expression that evaluates to True, False, or Unknown
- Use the LIKE predicate with string patterns for the search

Example:

- WHERE <columnname> LIKE <string pattern>

```
WHERE firstname LIKE R%
```



Retrieving rows - using a String Pattern

```
db2 => select firstname from Author  
        WHERE firstname like 'R%'
```

Firstname

Raul

Rav

2 record(s) selected.

Retrieving rows - using a Range

```
db2 => select title, pages from Book  
        WHERE pages >= 290 AND pages <= 300
```

Title	Pages
Database Fundamentals	300
Getting started with DB2 App Dev	298

2 record(s) selected.

```
db2 => select title, pages from Book  
        WHERE pages between 290 and 300
```

Title	Pages
Database Fundamentals	300
Getting started with DB2 App Dev	298

2 record(s) selected.



Retrieving rows - using a Set of Values

```
db2 => select firstname, lastname, country  
from Author  
      WHERE country='AU' OR country='BR'
```

Firstname	Lastname	Country
Xiqiang	Ji	AU
Juliano	Martins	BR

2 record(s) selected.

```
db2 => select firstname, lastname, country  
from Author  
      WHERE country IN ('AU', 'BR')
```

Firstname	Lastname	Country
Xiqiang	Ji	AU
Juliano	Martins	BR

2 record(s) selected.

Summary

Now you can describe how to simplify a SELECT statement by using:

- String patterns
- Ranges, or
- Sets of values

Sorting Result Sets

Sorting Result Sets

- At the end of this lesson, you will be able to:
 - Describe how to sort the result set by either ascending or descending order
 - Explain how to indicate which column to use for the sorting order

Sorting the Result Set

```
db2 => select * from Book
```

Book_ID	Title	Edition	Year	Price	ISBN	Pages	Aisle	Description
B1	Getting started with DB2 Express-C	1	2010	24.99	978-0-980046283-1-1	300	DB-A02	Teaches you the fundamentals
B2	Database Fundamentals	1	2010	24.99	978-0-986646283-5-1	280	DB-A01	Teaches you the essentials of
B3	Getting started with DB2 App Dev	1	2011	35.99	978-0-98086283-4-1	345	DB-A03	Teaches you the essentials of
B4	Getting started with WAS CE	1	2010	49.99	978-0-98946283-3-1	458	DB-A04	Teaches you the essentials of

4 record(s) selected.

```
db2 => select title from Book
```

Title

Getting started with DB2 Express-C

Database Fundamentals

Getting started with DB2 App Dev

Getting started with WAS CE

4 record(s) selected.

Using the ORDER BY clause

```
db2 => select title from Book
```

Title

Getting started with DB2 Express-C

Database Fundamentals

Getting started with DB2 App Dev

Getting started with WAS CE

4 record(s) selected.

```
db2 => select title from Book  
          ORDER BY title
```

Title

Database Fundamentals

Getting started with DB2 App Dev

Getting started with DB2 Express-C

Getting started with WAS CE

4 record(s) selected.

By default the result set is sorted in ascending order



ORDER BY clause - Descending order

```
db2 => select title from Book  
        ORDER BY title
```

Title

Database Fundamentals

Getting started with DB2 App Dev

Getting started with DB2 Express-C

Getting started with WAS CE

4 record(s) selected.

Ascending order by default

```
db2 => select title from Book  
        ORDER BY title DESC
```

Title

Getting started with WAS CE

Getting started with DB2 Express-C

Getting started with App Dev

Database Fundamentals

4 record(s) selected.

Descending order with DESC keyword



Specifying Column Sequence Number

```
db2 => select title, pages from Book  
        ORDER BY 2
```

Title	Pages
Getting started with WAS CE	278
Getting started with DB2 Express-C	280
Getting started with App Dev	298
Database Fundamentals	300

4 record(s) selected.

Ascending order by Column 2 (number of pages)



Summary

- Describe how to sort the result set by either ascending or descending order
- Explain how to indicate which column to use for the sorting order



Grouping Result Sets



Grouping Result Sets

- At the end of this lesson, you will be able to:
 - Eliminate duplicates from a result set
 - Describe how to further restrict a result set



Eliminating Duplicates - DISTINCT clause

```
db2 => select country from Author  
        ORDER BY 1
```

Country

AU

BR

...

CN

CN

...

IN

IN

IN

...

RO

RO

20 record(s) selected.

```
db2 => select distinct(country)  
        from Author
```

Country

AU

BR

CA

CN

IN

RO

6 record(s) selected.



GROUP BY clause

```
db2 => select country from Author  
        ORDER BY 1
```

Country

AU

BR

...

CN

CN

...

IN

IN

IN

...

RO

RO

20 record(s) selected.

```
db2 => select country, count(country)  
        from Author GROUP BY country
```

Country	2
AU	1
BR	1
CA	3
CN	6
IN	6
RO	3

6 record(s) selected.



GROUP BY clause

```
db2 => select country from Author  
        ORDER BY 1
```

Country

AU

BR

...

CN

CN

...

IN

IN

IN

...

RO

RO

20 record(s) selected.

```
db2 => select country, count(country)  
        as Count from Author group by country
```

Country	Count
AU	1
BR	1
CA	3
CN	6
IN	6
RO	3

6 record(s) selected.

Restricting the Result Set - HAVING clause

```
db2 => select country, count(country)
      as Count from Author group by country
```

Country	Count
AU	1
BR	1
CA	3
CN	6
IN	6
RO	3

6 record(s) selected.

```
db2 => select country, count(country)
      as Count from Author
      group by country
      having count(country) > 4
```

Country	Count
CN	6
IN	6

6 record(s) selected.

Summary

- Now you can:
 - Eliminate duplicates from a result set
 - Describe how to further restrict a result set



Search in course

Search



Tushar Raha

Databases and SQL for Data Science > Week 3 > Summary & Highlights

< Previous Next >

Refining your Results

✓ Video: Using String Patterns and Ranges
4 min

✓ Video: Sorting Result Sets
2 min

✓ Video: Grouping Result Sets
3 min

✓ Ungraded Plugin: Hands-on Lab : String Patterns, Sorting & Grouping
35 min

ⓘ Ungraded External Tool:
(Optional) Hands-on Lab: String Patterns, Sorting and Grouping
1h

ⓘ Reading: Summary & Highlights
5 min

ⓘ Ungraded Plugin: SQL Cheat Sheet: Intermediate - LIKE, ORDER BY, GROUP BY
15 min

Summary & Highlights

Congratulations! You have completed this lesson. At this point in the course, you know:

- You can use the WHERE clause to refine your query results.
- You can use the wildcard character (%) as a substitute for unknown characters in a pattern.
- You can use BETWEEN ... AND ... to specify a range of numbers.
- You can sort query results into ascending or descending order, using the ORDER BY clause to specify the column to sort on.
- You can group query results by using the GROUP BY clause.



[Back](#)

Practice Quiz

Practice Quiz • 15 min • 5 total points

Due Apr 9, 11:59 PM IST

1. You want to retrieve a list of employees in alphabetical order of Lastname from the Employees table. Which SQL statement should you use?

1 / 1 point

- SELECT * FROM Employees GROUP BY Lastname;
- SELECT * FROM Employees ORDER BY Lastname DESC;
- SELECT * FROM Employees ORDER BY Lastname;
- SELECT * FROM Employees SORT BY Lastname;



Correct. This SQL statement will retrieve a list of employees in alphabetical order from the Employees table.

2. Which keyword is used to set a condition for a GROUP BY clause?

1 / 1 point

- HAVING
- WHERE
- SELECT
- ORDER BY



Correct. The keyword HAVING is used to set a condition for a GROUP BY clause.

[Back](#)

Practice Quiz

Practice Quiz • 15 min • 5 total points

Due Apr 9, 11:59 PM IST

 ORDER BY **Correct**

Correct. The keyword HAVING is used to set a condition for a GROUP BY clause.

3. You want to retrieve a list of authors from Australia, Canada, and India from the table Authors. Which SQL statement is correct?

1 / 1 point

- SELECT * FROM Author IF Country ('Australia', 'Canada', 'India');
- SELECT * FROM Author WHERE Country BETWEEN('Australia', 'Canada', 'India');
- SELECT * FROM Author WHERE Country IN ('Australia', 'Canada', 'India');
- SELECT * FROM Author WHERE Country LIST ('CA', 'IN');

Correct

Correct. The IN keyword allows you to specify a list of values to match a condition.

4. You want to retrieve a list of books priced above \$10 and below \$25 from the table Book. What are the two ways you can specify the range?

1 / 1 point

- SELECT Title, Price FROM Book WHERE Price BETWEEN 10 and 25;

[Back](#) Practice Quiz

Practice Quiz • 15 min • 5 total points

Due Apr 9, 11:59 PM IST

Correct

Correct. The IN keyword allows you to specify a list of values to match a condition.

4. You want to retrieve a list of books priced above \$10 and below \$25 from the table Book. What are the two ways you can specify the range?

1 / 1 point

SELECT Title, Price FROM Book WHERE Price BETWEEN 10 and 25;

Correct

Partially correct. You can specify the price range using BETWEEN ... AND If you only selected this option, note that one other answer is also correct.

SELECT Title, Price FROM Book WHERE Price 10 to 25;

SELECT Title, Price FROM Book WHERE Price IN (10, 25);

SELECT Title, Price FROM Book WHERE Price >= 10 and Price <= 25;

Correct

Partially correct. You can specify the price range using the >= and <= operands. If you only selected this option, note that one other answer is also correct.

5. You want to retrieve Salary information for an employee called Ed from the Employee table. You write the following statement:

1 / 1 point

SELECT Firstname, Lastname, Salary FROM Employees

[Back](#)

Practice Quiz

Practice Quiz • 15 min • 5 total points

Due Apr 9, 11:59 PM IST

- SELECT Title, Price FROM Book WHERE Price >= 10 and Price <= 25;



Correct

Partially correct. You can specify the price range using the >= and <= operands. If you only selected this option, note that one other answer is also correct.

5. You want to retrieve Salary information for an employee called Ed from the Employee table. You write the following statement:

1 / 1 point

SELECT Firstname, Lastname, Salary FROM Employees

You see all the employees listed, and it's hard to find Ed's information. Which clause should you add to reduce the number of rows returned?

- ORDER BY Firstname;
- GROUP BY Firstname = 'Ed';
- WHERE Firstname = 'Ed';
- WHERE Employees = 'Ed';



Correct

Correct. The WHERE clause restricts the result set, in this case to employees with the first name Ed.

[Back](#)

Graded Quiz: Refining Your Results

Graded Quiz • 9 min

Due Apr 9, 11:59 PM IST

1. You want to select author's last name from a table, but you only remember the author's last name starts with the letter B, which string pattern can you use? 1 / 1 point

- SELECT lastname from author where lastname like 'B#'
- SELECT lastname from author where lastname like 'B%'
- SELECT lastname from author where lastname like 'B\$'
- None of the above



Correct. You can use the % sign as a wildcard to indicate zero or more missing characters.

2. In a SELECT statement, which SQL clause controls how the result set is displayed? 1 / 1 point

- ORDER BY clause
- ORDER IN clause
- ORDER WITH clause



Correct. You use the ORDER BY clause to control the order of rows in the result set.

[Back](#) Graded Quiz: Refining Your Results

Due Apr 9, 11:59 PM IST

Graded Quiz • 9 min

2. In a SELECT statement, which SQL clause controls how the result set is displayed?

1 / 1 point

- ORDER BY clause
- ORDER IN clause
- ORDER WITH clause



Correct. You use the ORDER BY clause to control the order of rows in the result set.

3. Which of the following can be used in a SELECT statement to restrict a result set?

1 / 1 point

- HAVING
- WHERE
- DISTINCT
- All of the above



Correct. You can use all of these keywords to restrict a result set.

Built-in Database Functions

Rav Ahuja

© IBM Corporation. All rights reserved.

Built-in Functions

- Most databases come with built-in SQL functions
- Built-in functions can be included as part of SQL statements
- Database functions can significantly reduce the amount of data that needs to be retrieved
- Can speed up data processing

PETRESUE TABLE

PETRESUE

ID INTEGER	ANIMAL VARCHAR(20)	QUANTITY INTEGER	COST DECIMAL(6,2)	RESCUEDATE DATE
1	Cat	9	450.09	2018-05-29
2	Dog	3	666.66	2018-06-01
3	Dog	1	100.00	2018-06-04
4	Parrot	2	50.00	2018-06-04
5	Dog	1	75.75	2018-06-10
6	Hamster	6	60.60	2018-06-11
7	Cat	1	44.44	2018-06-11
8	Goldfish	24	484.48	2018-06-14
9	Dog	2	222.22	2018-06-15

Aggregate or Column Functions

- INPUT: Collection of values (e.g. entire column)
- Output: Single value
- Examples: SUM(), MIN(), MAX(), AVG(), etc.

SUM

SUM function: Add up all the values in a column

```
SUM(COLUMN_NAME)
```

Example 1: Add all values in the COST column:

```
select SUM(COST) from PETRESCUE
```

Example 1: Result:

1

1718.24

Column Alias

Example 2: Explicitly name the output column SUM_OF_COST:

```
select SUM(COST) as SUM_OF_COST  
      from PETRESCUE
```

Example 2: Results:

SUM_OF_COST

1718.24

MIN, MAX

MIN: Return the MINIMUM value

MAX: Return the MAXIMUM value

Example 3A. Get the maximum QUANTITY of any ANIMAL:

```
select MAX (QUANTITY) from PETRESCUE
```

Example 3B. Results:

```
1  
24
```

MIN, MAX

Example 3B. Get the minimum value of ID column for Dogs:

```
select MIN(ID) from PETRESUE where ANIMAL = 'Dog'
```

Example 3B. Results:

1

2

Average

AVG() return the average value

Example 4. Specify the Average value of COST:

```
select AVG(COST) from PETRESCUE
```

Example 4. Results:

1

190.915555555555555555555555555555

Average

Mathematical operations can be performed between columns.

Example 5. Calculate the average COST per 'Dog':

```
select AVG(COST / QUANTITY) from PETRESCUE  
where ANIMAL = 'Dog'
```

Example 5. Results:

1

127.270000000000000000000000000000

SCALAR and STRING FUNCTIONS

SCALAR: Perform operations on every input value

Examples: ROUND(), LENGTH(), UCASE, LCASE

Example 6: Round UP or
DOWN every value in COST:

```
select  
    ROUND(COST)  
from PETRESCUE
```

Example 6. Results:

1	450.00
	667.00
	100.00
	50.00
	76.00

SCALAR and STRING FUNCTIONS

SCALAR: Perform operations on every input value

Examples: ROUND(), LENGTH(), UCASE, LCASE

Example 7. Retrieve the length of each value in ANIMAL:

```
select  
    LENGTH (ANIMAL)  
from PETRESCUE
```

Example 7. Results:

```
1  
3  
3  
3  
6  
3
```

UCASE, LCASE

Example 8: Retrieve ANIMAL values in UPPERCASE:

```
select UCASE (ANIMAL) from PETRESCUE
```

Example 8: Results:

1	
CAT	
DOG	
DOG	
PARROT	
DOG	

UCASE, LCASE

Example 9: Use the function in a WHERE clause :

```
select * from PETRESCUE  
where LCASE(ANIMAL) = 'cat'
```

Example 9: Results:

ID	ANIMAL	QUANTITY	COST	DATE
1	Cat	9	450.09	2018-05-29
7	Cat	1	44.44	2018-06-11

UCASE, LCASE

Example 10: Use the DISTINCT() function to get unique values :

```
select DISTINCT(UCASE(ANIMAL)) from PETRESCUE
```

Example 10: Results:

1

CAT

DOG

GOLDFISH

HAMSTER

PARROT

Date and Time Built-in Functions

Rav Ahuja

Date, Time Functions

Most databases contain special datatypes for dates and times.

DATE: YYYYMMDD

TIME: HHMMSS

TIMESTAMP: YYYYXXDDHHMMSSZZZZZ

Date / Time functions:

YEAR(), MONTH(), DAY(), DAYOFMONTH(), DAYOFWEEK(),
DAYOFYEAR(), WEEK(), HOUR(), MINUTE(), SECOND()

Date, Time Functions (continued)

Example 11: Extract the DAY portion from a date:

```
select DAY(RESCUEDATE) from PETRESCUE  
where ANIMAL='Cat'
```

Example 11: Results:

ID	ANIMAL	QUANTITY	COST	RESCUEDATE
1	Cat	9	450.09	5/29/2018
7	Cat	1	44.44	6/11/2018

29 ←

11 ←

Date, Time Functions (continued)

Example 12: Get the number of rescues during the month of May :

```
select COUNT(*) from PETRESCUE  
where MONTH(RESCUEDATE) = '05'
```

Example 12: Results:

```
1
```

Date or Time Arithmetic

Example 13: What date is it 3 days after each rescue date?

Select (RESCUEDATE + 3 DAYS) from PETRESCUE

Example 13: Results:

	ID	ANIMAL	QUANTITY	COST	RESCUEDATE	+ 3 DAYS
2018-06-01		1Cat	9	450.09	5/29/2018	6/1/2018
2018-06-04		2Dog	3	666.66	6/1/2018	6/4/2018
2018-06-07		3Dog	1	100	6/4/2018	6/7/2018
2018-06-07		4Parrot	2	50	6/4/2018	6/7/2018
2018-06-13		5Dog	1	75.75	6/10/2018	6/13/2018

Date or Time Arithmetic

Special Registers:

CURRENT_DATE, CURRENT_TIME

Example 14: Find how many days have passed since each RESCUEDATE till now:

```
Select (CURRENT_DATE - RESCUEDATE) from PETRESCUE
```

Example 14: Sample result (format YMMDD):

10921

SUB-QUERIES and NESTED SELECTS

Rav Ahuja

© IBM Corporation. All rights reserved.

Sub-queries and Nested Selects

Sub-query: A query inside another query

```
select COLUMN1 from TABLE  
where COLUMN2 = (select MAX(COLUMN2) from TABLE)
```

EMPLOYEES

EMP_ID	F_NAME	L_NAME	SSN	B_DATE	SEX	ADDRESS	JOB_ID	SALARY	MANAGER_ID	DEP_ID
E1001	John	Thomas	123456	1976-01-09	M	5631 Rice, OakPark,IL	100	100000	30001	2
E1002	Alice	James	123457	1972-07-31	F	980 Berry In, Elgin,IL	200	80000	30002	5
E1003	Steve	Wells	123458	1980-08-10	M	291 Springs, Gary,IL	300	50000	30002	5

Why use sub-queries?

To retrieve the list of employees who earn more than the average salary:

```
select * from employees  
      where salary > AVG(salary)
```

This query will result in error:

SQL0120N Invalid use of an aggregate function or OLAP function. SQLCODE=-120, SQLSTATE=42903

Sub-queries to evaluate Aggregate functions

- Cannot evaluate Aggregate functions like AVG() in the WHERE clause –
- Therefore, use a sub-Select expression:

```
select EMP_ID, F_NAME, L_NAME, SALARY  
      from employees  
     where SALARY <  
           (select AVG(SALARY) from employees);
```

Sub-queries to evaluate Aggregate functions

Result:

EMP_ID	F_NAME	L_NAME	SALARY
E1003	Steve	Wells	50000.00
E1004	Santosh	Kumar	60000.00
E1007	Mary	Thomas	65000.00

Sub-queries in list of columns

- Substitute column name with a sub-query
- Called Column Expressions

```
select EMP_ID, SALARY, AVG(SALARY) AS AVG_SALARY  
      from employees ;
```

```
select EMP_ID, SALARY,  
      ( select AVG(SALARY) from employees )  
            AS AVG_SALARY  
      from employees ;
```

Sub-queries in list of columns

Result:

EMP_ID	SALARY	AVG_SALARY
E1002	80000.00	68888.88888888888888888888888888
E1003	50000.00	68888.88888888888888888888888888
E1004	60000.00	68888.88888888888888888888888888
E1005	70000.00	68888.88888888888888888888888888
E1006	90000.00	68888.88888888888888888888888888
E1007	65000.00	68888.88888888888888888888888888
E1008	65000.00	68888.88888888888888888888888888
E1009	70000.00	68888.88888888888888888888888888
E1010	70000.00	68888.88888888888888888888888888

Sub-queries in FROM clause

- Substitute the TABLE name with a sub-query
- Called Derived Tables or Table Expressions
- Example:

```
select * from  
  ( select EMP_ID, F_NAME, L_NAME, DEP_ID  
    from employees) AS EMP4ALL ;
```

Sub-queries in FROM clause

Result:

EMP_ID	F_NAME	L_NAME	DEP_ID
E1002	Alice	James	5
E1003	Steve	Wells	5
E1004	Santosh	Kumar	5
E1005	Ahmed	Hussain	2
E1006	Nancy	Allen	2
E1007	Mary	Thomas	7
E1008	Bharath	Gupta	7
E1009	Andrea	Jones	7
E1010	Ann	Jacob	5

Summary

In this video you have learned:

- How sub-queries and nested queries form richer queries
- How they overcome limitations of aggregate functions
- How to use sub-queries in the:
 - WHERE clause
 - list of columns
 - FROM clause

QUERYING MULTIPLE TABLES

Rav Ahuja

© IBM Corporation. All rights reserved.

Working with Multiple Tables

Ways to access multiple tables in the same query:

1. Sub-queries
2. Implicit JOIN
3. JOIN operators (INNER JOIN, OUTER JOIN, etc.)

Tables for Examples in this Lesson

EMPLOYEES:

EMP_ID	F_NAME	L_NAME	SSN	B_DATE	SEX	ADDRESS	JOB_ID	SALARY	MANAGER_ID	DEP_ID
E1001	John	Thomas	123456	1976-01-09	M	5631 Rice, OakPark,IL	100	100000	30001	2
E1002	Alice	James	123457	1972-07-31	F	980 Berry In, Elgin,IL	200	80000	30002	5
E1003	Steve	Wells	123458	1980-08-10	M	291 Springs, Gary,IL	300	50000	30002	5

DEPARTMENTS:

DEPT_ID_DEP	DEP_NAME	MANAGER_ID	LOC_ID
5	Software Development	30002	L0002
7	Design Team	30003	L0003

Accessing Multiple Tables with Sub-queries

To retrieve only the employee records that correspond to departments in the DEPARTMENTS table:

```
select * from employees  
      where DEP_ID IN  
        ( select DEPT_ID_DEP from departments );
```

Accessing Multiple Tables with Sub-queries

Result:

EMP_ID	F_NAME	L_NAME	SSN	B_DATE	SEX	ADDRESS	JOB_ID	SALARY	MANAGER_ID	DEP_ID
E1002	Alice	James	123457	7/31/1972	F	980 Berry Ln, Elgin,IL	200	80000	30002	5
E1003	Steve	Wells	123458	8/10/1980	M	291 Springs, Gary,IL	300	50000	30002	5
E1004	Santosh	Kumar	123459	7/20/1985	M	511 Aurora Av, Aurora,IL	400	60000	30004	5
E1007	Mary	Thomas	123412	5/5/1975	F	100 Rose Pl, Gary,IL	650	65000	30003	7
E1008	Bharath	Gupta	123413	5/6/1985	M	145 Berry Ln, Naperville,IL	660	65000	30003	7
E1009	Andrea	Jones	123414	7/9/1990	F	120 Fall Creek, Gary,IL	234	70000	30003	7
E1010	Ann	Jacob	123415	3/30/1982	F	111 Britany Springs,Elgin,IL	220	70000	30004	5

Multiple Tables with Sub-queries

To retrieve only the list of employees from a specific location:

- EMPLOYEES table does not contain location information
- Need to get location info from DEPARTMENTS table

```
select * from employees  
      where DEP_ID IN  
        ( select DEPT_ID_DEP from departments  
          where LOC_ID = 'L0002' );
```

Multiple Tables with Sub-queries

Result:

EMP_ID	F_NAME	L_NAME	SSN	B_DATE	SEX	ADDRESS	JOB_ID	SALARY	MANAGER_ID	DEP_ID
E1002	Alice	James	123457	7/31/1972	F	980 Berry ln, Elgin,IL	200	80000	30002	5
E1003	Steve	Wells	123458	8/10/1980	M	291 Springs, Gary,IL	300	50000	30002	5
E1004	Santosh	Kumar	123459	7/20/1985	M	511 Aurora Av, Aurora,IL	400	60000	30004	5
E1010	Ann	Jacob	123415	3/30/1982	F	111 Britany Springs,Elgin,IL	220	70000	30004	5

Multiple Tables with Sub-queries

To retrieve the department ID and name for employees who earn more than \$70,000:

```
select DEPT_ID_DEP, DEP_NAME from departments  
where DEPT_ID_DEP IN  
( select DEP_ID from employees  
where SALARY > 70000 ) ;
```

Multiple Tables with Sub-queries

Result:

DEPT_ID	DEP_NAME
5	Software Group

Accessing multiple tables with Implicit Join

Specify 2 tables in the FROM clause:

```
select * from employees, departments;
```

The result is a full join (or Cartesian join):

- Every row in the first table is joined with every row in the second table
- The result set will have more rows than in both tables

Accessing multiple tables with Implicit Join

Result:

EMP_ID	F_NAME	L_NAME	SSN	B_DATE	SEX	ADDRESS	JOB_ID	SALARY	MANAGER_ID	DEP_ID	DEPT_ID	DEP	DEP_NAME	MANAGER_ID	LOC_ID
E1002	Alice	James	123457	7/31/1972F		980 Berry ln, Elgin,IL	200	80000	30002	5		Software	5Group		30002L0002
E1003	Steve	Wells	123458	8/10/1980M		291 Springs, Gary,IL	300	50000	30002	5		Software	5Group		30002L0002
E1004	Santosh	Kumar	123459	7/20/1985M		511 Aurora Av, Aurora,IL	400	60000	30002	5		Software	5Group		30002L0002
E1005	Ahmed	Hussain	123410	1/4/1981M		216 Oak Tree, Geneva,IL	500	70000	30002	2		Software	5Group		30002L0002
E1006	Nancy	Allen	123411	2/6/1978F		111 Green Pl, Elgin,IL	600	90000	30002	2		Software	5Group		30002L0002
E1007	Ahmed	1/4/1981M				70000				/Team			
E1006	Nancy	Allen	123411	2/6/1978F		111 Green Pl, Elgin,IL	600	90000	30003	2		Design	7Team		30003L0003
E1007	Mary	Thomas	123412	5/5/1975F		100 Rose Pl, Gary,IL	650	65000	30003	7		Design	7Team		30003L0003
E1008	Bharath	Gupta	123413	5/6/1985M		145 Berry Ln, Naperville,IL	660	65000	30003	7		Design	7Team		30003L0003
E1009	Andrea	Jones	123414	7/9/1990F		120 Fall Creek, Gary,IL	234	70000	30003	7		Design	7Team		30003L0003
E1010	Ann	Jacob	123415	3/30/1982F		111 Britany Springs,Elgin,IL	220	70000	30003	5		Design	7Team		30003L0003

Accessing multiple tables with Implicit Join

Use additional operands to limit the result set:

```
select * from employees, departments  
      where employees.DEP_ID =  
            departments.DEPT_ID_DEP;
```

Use shorter aliases for table names:

```
select * from employees E, departments D  
      where E.DEP_ID = D.DEPT_ID_DEP;
```

Accessing multiple tables with Implicit Join

Query:

```
select * from employees E, departments D  
      where E.DEP_ID = D.DEPT_ID_DEP;
```

Result:

EMP_ID	F_NAME	L_NAME	SSN	B_DATE	SEX	ADDRESS	JOB_ID	SALARY	MANAGER_ID	DEP_ID	DEPT_ID	DEP_NAME	MANAGER_ID	LOC_ID
E1002	Alice	James	123457	7/31/1972	F	980 Berry In, Elgin,IL	200	80000	30002	5		Software S Group	30002	L0002
E1003	Steve	Wells	123458	8/10/1980	M	291 Springs, Gary,IL	300	50000	30002	5		Software S Group	30002	L0002
E1004	Santosh	Kumar	123459	7/20/1985	M	511 Aurora Av, Aurora,IL	400	60000	30002	5		Software S Group	30002	L0002
E1007	Mary	Thomas	123412	5/5/1975	F	100 Rose Pl, Gary,IL	650	65000	30003	7		7 Design Team	30003	L0003
E1008	Bharath	Gupta	123413	5/6/1985	M	145 Berry Ln, Naperville,IL	660	65000	30003	7		7 Design Team	30003	L0003
E1009	Andrea	Jones	123414	7/9/1990	F	120 Fall Creek, Gary,IL	234	70000	30003	7		7 Design Team	30003	L0003
E1010	Ann	Jacob	123415	3/30/1982	F	111 Britany Springs,Elgin,IL	220	70000	30002	5		Software S Group	30002	L0002

Accessing multiple tables with Implicit Join

To see the department name for each employee:

```
select EMP_ID, DEP_NAME  
      from employees E, departments D  
     where E.DEP_ID = D.DEPT_ID_DEP;
```

Accessing multiple tables with Implicit Join

Query:

```
select EMP_ID, DEP_NAME from employees E, departments D  
      where E.DEP_ID = D.DEPT_ID_DEP;
```

Result:

EMP_ID	DEP_NAME
E1002	Software Group
E1003	Software Group
E1004	Software Group
E1007	Design Team
E1008	Design Team
E1009	Design Team
E1010	Software Group

Accessing multiple tables with Implicit Join

Column names in the select clause can be pre-fixed by aliases:

```
select E.EMP_ID, D.DEP_ID_DEP from  
      employees E, departments D  
  where E.DEP_ID = D.DEPT_ID_DEP
```

Accessing multiple tables with Implicit Join

Query:

```
select E.EMP_ID, D.DEPARTMENT_ID DEP from employees E, departments D  
      where E.DEP_ID = D.DEPARTMENT_ID DEP;
```

Result:

EMP_ID	DEPARTMENT_ID DEP
E1002	5
E1003	5
E1004	5
E1005	2
E1006	2
E1007	7
E1008	7
E1009	7
E1010	5



Search in course

Search



Tushar Raha

Databases and SQL for Data Sci...

> Week 3

> Summary & Highlights

< Previous Next >

Working with Multiple Tables
queries and Nested SELECTS
1h

Video: Working with Multiple Tables
6 min

Ungraded Plugin: Hands-on Lab: Working with Multiple Tables
30 min

Ungraded External Tool:
(Optional) Hands-on Lab: Working with Multiple Tables
1h

Reading: Summary & Highlights
5 min

Ungraded Plugin: SQL Cheat Sheet: FUNCTIONS and Implicit JOIN
15 min

Practice Quiz: Practice Quiz
5 questions

Summary & Highlights

Congratulations! You have completed this lesson. At this point in the course, you know:

- Most databases come with built-in functions that you can use in SQL statements to perform operations on data within the database itself.
- When you work with large datasets, you may save time by using built-in functions rather than first retrieving the data into your application and then executing functions on the retrieved data.
- You can use sub-queries to form more powerful queries than otherwise.
- You can use a sub-select expression to evaluate some built-in aggregate functions like the average function.
- Derived tables or table expressions are sub-queries where the outer query uses the results of the sub-query as a data source.

Mark as completed



[Back](#) Practice Quiz

Practice Quiz • 15 min • 5 total points

Due Apr 9, 11:59 PM IST

1. Which of the following statements about built-in database functions is correct?

1 / 1 point

- Built-in database functions reduce the amount of data that is retrieved.
- Built-in database functions must be called from a programming language like Python.
- Built-in database functions may increase processing time.
- Built-in database functions may increase network bandwidth consumed.

 Correct

Correct. Built-in database functions process within the database itself, so the amount of data that is retrieved to the client machine is significantly reduced.

2. Which of the following SQL queries would return the day of the week each dog was rescued?

1 / 1 point

- SELECT RescueDate From PetRescue WHERE Animal = 'Dog';
- SELECT DAY(RescueDate) From PetRescue WHERE Animal = 'Dog';
- SELECT DAYOFWEEK(RescueDate) From PetRescue WHERE Animal = 'Dog';
- SELECT DAYOFWEEK(RescueDate) From PetRescue;

 Correct

Correct. The DAYOFWEEK() function returns the day of the week, and the WHERE clause correctly specifies the animal as a dog.

[Back](#) Practice Quiz

Practice Quiz • 15 min • 5 total points

Due Apr 9, 11:59 PM IST

3. What is the result of the following query: **SELECT (Current_Date - RescueDate) FROM PetRescue**

1 / 1 point

- Returns how long it has been since each rescue.
- Returns the rescue date for each rescue.
- Returns today's date.
- Returns the current date and rescue date columns.



Correct. This query returns how long it has been since the rescue.

4. Which of the following queries will return the employees who earn less than the average salary?

1 / 1 point

- SELECT * FROM Employees WHERE Salary < AVG(Salary)
- SELECT * FROM Employees WHERE Salary < (SELECT AVG(Salary) FROM Employees);
- SELECT AVG(Salary) FROM Employees WHERE Salary < AVG(Salary)
- SELECT * FROM Employees WHERE Salary < (SELECT AVG(Salary))



Correct. The AVG(Salary) function must be included in a sub-query within the WHERE clause.

[Back](#) Practice Quiz

Practice Quiz • 15 min • 5 total points

Due Apr 9, 11:59 PM IST

- SELECT * FROM Employees WHERE Salary < (SELECT AVG(Salary) FROM Employees);
- SELECT AVG(Salary) FROM Employees WHERE Salary < AVG(Salary)
- SELECT * FROM Employees WHERE Salary < (SELECT AVG(Salary))



Correct. The AVG(Salary) function must be included in a sub-query within the WHERE clause.

5. What are the three ways to work with multiple tables in the same query?

1 / 1 point

- Built-in functions, implicit joins, JOIN operators
- Sub-queries, Implicit joins, normalization.
- Sub-queries, APPEND, JOIN operators
- Sub-queries, Implicit joins, JOIN operators



Correct: You can retrieve information from more than one table by using a sub-query, an implicit join, or a JOIN operator like INNER JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN, or FULL OUTER JOIN.